# 正则表达式

## 什么是正则表达式

正则表达式(regular expression)，就是一个规则，用来匹配字符串中的字符

## 创建正则表达式

### new方式

let regExp3 = new RegExp('box', 'i');  
let regExp4 = new RegExp('\\d'); // 反斜杠 \ 在字符串中代表转义，必须用双反斜杠

### 字面量方式

写在两个正斜杠之间  
let regExp1 = /box/i; // 匹配一个0 - 9之间的数字  
let regExp2 = /\d/; // 匹配 box 字符串，不区分大小写

### 修饰符

i : 忽略大小写； g：全局匹配 ；

## 使用正则表达式

### 正则表达式对象的方法

test() 测试字符串和正则表达式是否匹配，返回布尔值  
exec() 测试字符串是否和正则表达式是否匹配，返回结果数组或null

正则表达式对象有两个方法 test() 和 exec()

### 字符串的方法

search(reg) 返回匹配到的字符串下标，没有匹配到则返回-1

match(reg) 以数组形式返回匹配到的字符串，没有匹配到则返回null

replace(reg,'content') 按照正则表达式进行替换，返回一个字符串

split(reg) 以正则表达式为分隔符对字符串进行分割，返回数组

## 元字符

\d 匹配一个数字 等效于 [0-9]  
\D 匹配一个非数字 等效于 [^0-9]  
\w 匹配数字、字母及下划线 [0-9a-zA-Z\_]  
\W 匹配非数字、字母及下划线 [^\w] 或 [^0-9a-zA-Z\_]  
\s 匹配任何空白字符：空格、制表符、换行  
\S 匹配任何非空白字符

## 限定符

字符 描述

^ 行首匹配  
$ 行尾匹配  
\* 匹配前面的子表达式零次或多次  
+ 匹配前面的子表达式一次或多次  
? 匹配前面的子表达式零次或一次  
. 匹配任意字符，除了换行  
{n} 花括号：限定数量，匹配确定的 n 次  
{n,} 至少匹配n 次  
{n,m} 最少匹配 n 次且最多匹配 m 次  
[] 限定范围，如[0-9] ，[a-d] , [abc]  
() 分组  
| 或

## 示例

### 判断一个字符串是不是由数字组成

### 找出连续的多个数字

要求结果：*["1", "23", "456", "7890"]*

<**script**>  
 **let *str*** = **'abc1de23fg456hijk7890lmn'**;  
</**script**>

要求结果：*["<123>", "<33>"]*

<**script**>  
 **let *str*** = **'abc<123>fdfd<33>fff34fdf<>'**;  
</**script**>

### 正则验证QQ号

5-10个位数字；第一个数字不能是0

### 删除空格

删除首空格：

删除尾空格：

删除所有空格：

### 是否是压缩文件

zip,rar,gz

### 正则验证身份证号码

可以是15位或者是18位，其中最后一位可以是X，其它全是数字，

可以是 不代表 必须是； 最后一位也许是X也许是数字

### 取出字符串最后一组数字

如

30CAC0040 取出0040

3SFASDF92 取出92

### 使用场景

我想知道邮箱格式、手机格式、IP格式合不合法，用test  
如果想抓取网页中所有的手机号，使用exec或match  
想替换掉网页中敏感词汇，用replace